# PCS 2428 / PCS 2059
# Inteligência Artificial

Prof. Dr. Jaime Simão Sichman
Prof. Dra. Anna Helena Reali Costa

Lógica de Predicados

## Representação de Conhecimento

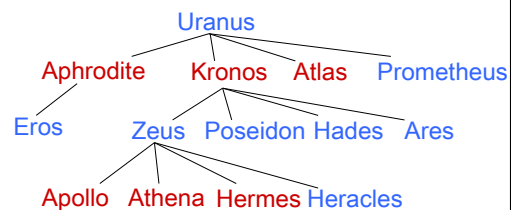
- Abordagem procedural
  - Mundo do Wumpus em matriz (4,4)
  - Se existir um buraco em [2,2], coloco um valor “B” na posição correspondente da matriz
  - Não tem um mecanismo geral para derivar fatos de outros fatos, isto é feito por procedimentos dependentes do domínio que o programador cria!
- Abordagem declarativa
  - Conhecimento e inferência são separados
  - Inferência é independente do domínio!
  - Linguagens lógicas também são composicionais
    - Semântica de uma sentença pode ser derivada de suas partes

## Cálculo Proposicional: Limitações

Apesar de ser simples, o cálculo proposicional não consegue expressar fatos genéricos de modo conciso, como por exemplo “se o agente estiver em qualquer local com o Wumpus à frente, não prosseguir em frente”.

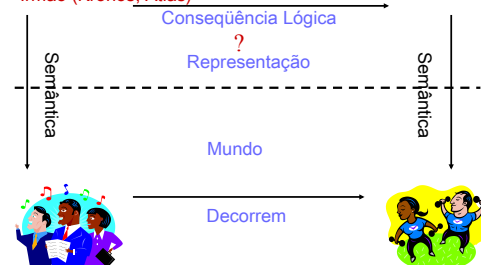
Seria interessante poder expressar tais fatos genéricos através de um linguagem que pudesse fazer referência a objetos e a algumas de suas propriedades.

## Cálculo Proposicional: Limitações

Exemplo - Família dos Deuses Gregos

## Cálculo de Predicados: Definições

O cálculo de predicados permite representar:

- **Objetos**
  - exemplo: Aphrodite e Kronos
- **Relações** entre objetos
  - exemplo: Aphrodite é irmã de Kronos
- **Funções** de objetos
  - exemplo: Uranus é pai de Aphrodite

## Cálculo de Predicados: Definições

## Comparação entre Linguagens Lógicas

- Linguagens lógicas têm compromissos:
  - Ontológicos: como representar o mundo
  - Epistemológicos: como um agente crê nestes fatos

| Linguagem | Compromisso Ontológico | Compromisso Epistemológico |
|---|---|---|
| L. Proposicional | fatos | verdadeiro, falso, desconhecido |
| L. Predicados | fatos, objetos, relações | verdadeiro, falso, desconhecido |
| L. Temporal | fatos, objetos, relações, tempos | verdadeiro, falso, desconhecido |
| T. Probabilidades | fatos | graus de crença ∈ [0,1] |
| L. Nebulosa | fatos, graus de crença ∈ [0,1] | intervalos de valores definidos |

## Sintaxe do Cálculo de Predicados: Símbolos

Os elementos básicos da sintaxe do cálculo de predicados são **símbolos** que se referem a objetos, relações e funções:

- símbolos para **constantes**, que se referem a objetos, como Aphrodite e Zeus;
- símbolos para **variáveis,** que se referem a objetos ou conjunto de objetos, como x, y, z, ….;
- símbolos para **predicados**, que se referem a relações entre objetos, como Irmão, Primo;
- símbolos para **funções**, que se referem a funções de objetos, como Pai, Mãe.

## Sintaxe do Cálculo de Predicados : Termos

**Termos** são expressões lógicas que se referem a um objeto.

Termos podem ser assim constituídos por:

- constantes, como Aphrodite e Zeus;
- variáveis**,** como x, y, z, ….;
- funções aplicadas a termos, como Pai(Aphrodite), Mãe(z), Mãe(Pai(Zeus)).

## Sintaxe do Cálculo de Predicados: Predicados

Uma **sentença atômica** será composta por um predicado, aplicado a um ou mais termos, dependendo de sua aridade:

- Irmão (Kronos, Aphrodite)
- Irmão (Pai (Zeus), Mãe (Eros))
- Irmão (x, y)

**Fórmulas bem formadas** serão compostas por predicados ligados através de conectivos lógicos:

- Irmão(Pai(Zeus), Mãe(Eros))→Primo(Zeus, Eros)

## Sintaxe do Cálculo de Predicados em BNF

*Símbolo inicial:* <sentença>
<sentença>::= <sentença atômica> | ¬<sentença>|
( <sentença> <conectivo> <sentença> ) |
<quantificador> <variável> <sentença>
<sentença atômica>::= <predicado> (<termo>, ...)
<termo> ::= <função> (<termo>, ...) | <constante> |
<variável>
<conectivo> ::= → | ∧ | ∨ | ↔
<quantificador> ::= ∀ | ∃
<constante> ::= A | $X_1$ | João | ...
<variável> ::= x | y | z | ...
<predicado> ::= Antes | TemCor | Chovendo | ...
<função> ::= Mãe | NUSP | ...

## Quantificadores

**Quantificador Universal**

*Sintaxe*: ∀ x P(x)

Lê-se para todo x, P(x) ou para qualquer x, P(x).

**Quantificador Existencial**

*Sintaxe*: ∃ x P(x)

Lê-se para algum x, P(x) ou existe x, P(x).

## Quantificadores

Exemplos de expressões usando quantificadores:

- Todo homem é mortal
  $\forall x\ (Homem(x) \rightarrow Mortal(x))$
- Nenhum homem é imortal
  $\forall x\ (Homem(x) \rightarrow \neg Imortal(x))$
- Pelo menos um homem é inteligente
  $\exists x\ (Homem(x) \wedge Inteligente\ (x))$


## Quantificadores

Exemplos de expressões usando quantificadores:

- Todo homem é mortal
  $\forall x\ (Homem(x) \rightarrow Mortal(x))$
- Nenhum homem é imortal
  $\forall x\ (Homem(x) \rightarrow \neg Imortal(x))$
- Pelo menos um homem é inteligente
  $\exists x\ (Homem(x) \wedge Inteligente\ (x))$


## Quantificadores

Exemplos de expressões usando quantificadores encadeados:

- Todo homem ama as mulheres
  $\forall x\ \forall y\ (Homem(x) \wedge Mulher(y) \rightarrow Ama(x, y))$
- Todo homem ama uma mulher
  $\forall x\ \exists y\ (Homem(x) \wedge Mulher(y) \rightarrow Ama(x, y))$
- Há uma mulher amada por todos os homens
  $\exists y\ \forall x\ (Homem(x) \wedge Mulher(y) \rightarrow Ama(x, y))$

## Quantificadores

**Teorema:**

Generalização da Lei de De Morgan para os quantificadores

$$\neg \forall x\ P(x) \equiv \exists x\ \neg\ P(x)$$

$$\neg \exists x\ P(x) \equiv \forall x\ \neg\ P(x)$$

Um exemplo intuitivo seria:

- “Não é todo homem que é egoísta” equivale a “Existe pelo menos um homem que não é egoísta”

## Quantificadores: Variáveis Livres e Ligadas

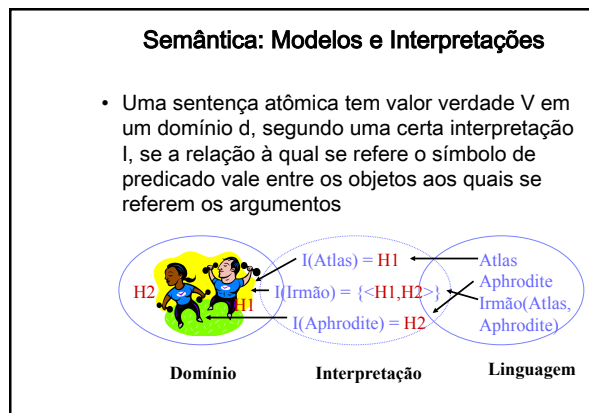
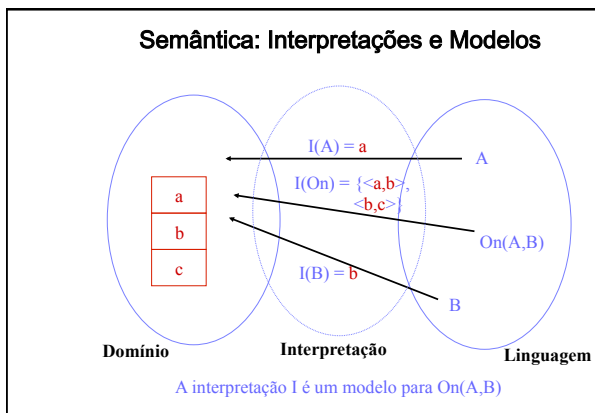
No cálculo de predicados, devido à presença dos quantificadores, uma variável pode estar:

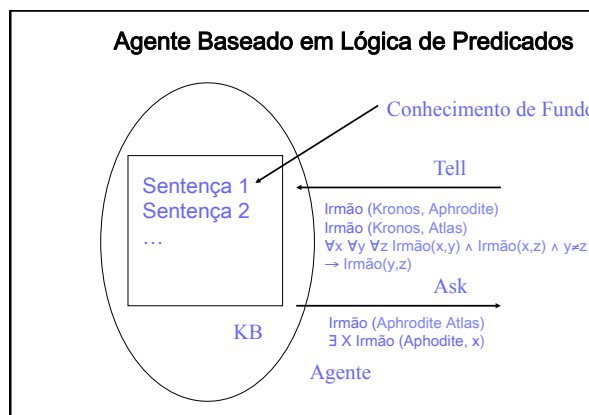
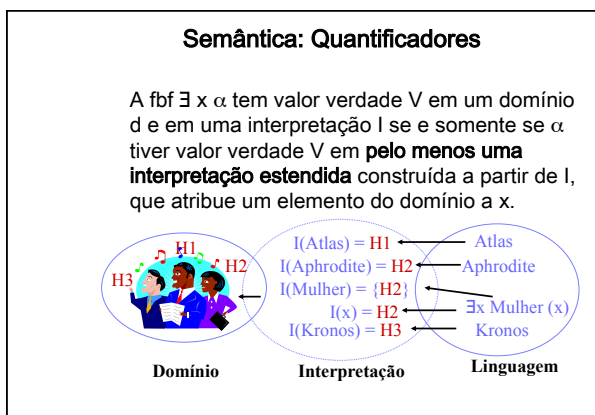
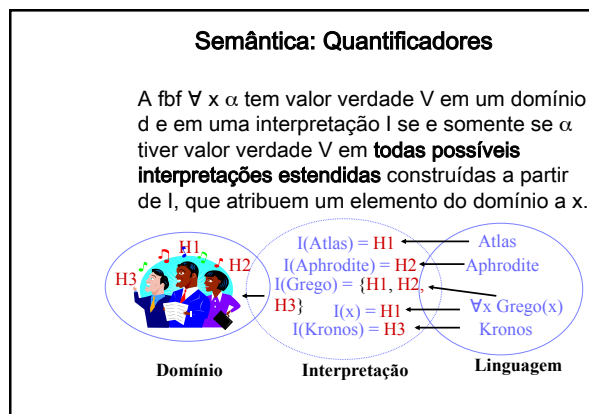
- **livre** (free)
  - $Irmão(x,y) \rightarrow \neg FilhoÚnico(x)$
- **ligada** (bounded)
  - $\forall x\ \forall y\ (Irmão(x,y) \rightarrow \neg FilhoÚnico(x))$

Estando ligada, diz-se que uma variável está no **escopo** de um quantificador.

## Quantificadores: Sentenças Abertas, Fechadas e Primitivas

No cálculo de predicados, uma sentença pode ser:

- **aberta:** sem restrições
  - $Irmão\ (x, y) \rightarrow MesmoPai\ (x, y)$
- **fechada** (closed): sem variáveis livres
  - $\exists x\ Irmão\ (Kronos, x)$
- **primitiva** (grounded): sem variáveis livres ou ligadas
  - $Irmão\ (Kronos, Aphrodite) \rightarrow MesmoPai\ (Kronos, Aphrodite)$

## Semântica: Fórmulas Bem Formadas

- A semântica das fórmulas bem formadas, construídas utilizando os conectivos lógicos, é definida do mesmo modo que no cálculo proposicional.
- De modo semelhante, fbfs podem ser válidas, inválidas, contingentes, satisfatíveis ou insatisfatíveis.
- A única diferença diz respeito à definição da semântica dos quantificadores.

## Inferência

- O procedimento de inferência em lógica de predicados é semelhante ao visto em lógica proposicional, porém mais complexo
  - presença dos quantificadores
  - presença de variáveis
- Este procedimento leva em conta duas noções fundamentais:
  - substituição
  - unificação

## Inferência: Substituição

Uma **substituição** é um conjunto finito de associações entre variáveis e expressões tais que:

- cada variável é associada no máximo com uma única expressão
- nenhuma variável com uma expressão associada ocorre no escopo de qualquer outra expressão
- ex: {x/A, y/F(B), z/w} é uma substituição
  {x/G(y), y/F(x)} não é uma substituição

## Inferência: Substituição

Uma substituição pode ser aplicada numa fbf do cálculo de predicados, gerando uma instância desta substituição.

A notação utilizada é $\phi\sigma$, onde se aplica a substituição $\sigma$ à expressão $\phi$

- ex: P(x, x, y, v) {x/A, y/F(B), z/w}
  resulta em:
  P(A, A, F(B), v)

## Inferência: Unificação

Trata-se do processo que determina se duas expressões podem se tornar idênticas se suas variáveis forem substituídas de modo apropriado.

Um conjunto de expressões $\{\phi_1, \phi_2, ..., \phi_n\}$ é **unificável** se e sómente se existir uma susbtituição $\sigma$ que as torna idênticas:

$\phi_1\sigma = \phi_2\sigma = ...= \phi_n\sigma$. Neste caso, $\sigma$ é dito um **unificador** do conjunto.

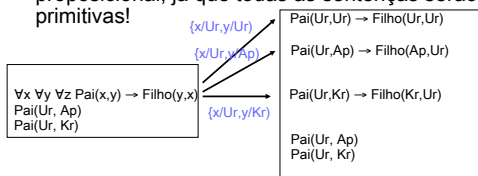
- ex: P(A, y, z) {x/A, y/B, z/C} = P(A, B, C) =
  P(x, B, z) {x/A, y/B, z/C}

## Regras de Inferência para Cálculo de Predicados

Instanciação Universal $\forall x\ \alpha \models \alpha\ \{x/g^1\}$
Instanciação Existencial $\exists x\ \alpha \models \alpha\ \{x/c^2\}$

1. O termo g deve ser um termo primitivo
2. A constante c não deve ter aparecido na BC

Com estas regras, pode-se construir uma BC proposicional, já que todas as sentenças serão primitivas!

Pai(Ur,Ur) → Filho(Ur,Ur)
Pai(Ur,Ap) → Filho(Ap,Ur)
Pai(Ur,Kr) → Filho(Kr,Ur)
Pai(Ur, Ap)
Pai(Ur, Kr)

## Regras de Inferência para Cálculo de Predicados

Modus Ponens Generalizado

$p1', p2', \ldots, pn', p1 \wedge p2 \wedge \ldots \wedge p1 \rightarrow q \models q\sigma$
onde $pi\ \sigma = pi'\ \sigma$, $i \in \{1..n\}$

- Mais eficiente, pois não preciso gerar sentenças desnecessárias
- Existem algoritmos eficientes para achar os unificadores. Em particular, existem infinitos unificadores para duas sentenças, e sempre busca-se o unificador mais geral (mgu)

## Exemplo : Modus Ponens Generalizado

Dado o domínio dos Deuses Gregos, considere as sentenças:

1. Uma pessoa é avô de outra se for pai de seu pai ou de sua mãe;
2. Kronos é pai de Zeus;
3. Uranus é pai de Kronos.

Provar que Uranus é avô de Zeus.

## Exemplo : Modus Ponens Generalizado

1. ∀x∀y∀z Pai(x,y) ∧ (Pai(y,z) ∨ Mãe(y,z)) → Avô (x, z) — premissa
2. Pai (Kronos, Zeus) — premissa
3. Pai (Uranus, Kronos) — premissa
4. Pai (Kronos, Zeus) ∨ Mãe(Kronos, Zeus) — AD 2
5. Pai (Uranus, Kronos) ∧ (Pai (Kronos, Zeus) ∨ Mãe(Kronos, Zeus)) — CJ 3, 4
6. Avô (Uranus, Zeus) — MPG 1,5

{x/Uranus, y/Kronos, z/Zeus}

## Transformação para Forma Clausal

Prova-se que qualquer fbf do cálculo de predicados pode ser transformada num conjunto de cláusulas equivalente, através de uma sequência definida de passos.

Exemplo: "Todo aquele que ama todos os animais é amado por alguém"

Como seria a representação disto em lógica de predicados?

∀x (∀y Animal(y) → Loves(x,y)) → (∃y Loves(y, x))

## Transformação para Forma Clausal

∀x (∀y Animal(y) → Loves(x,y)) → (∃y Loves(y, x))

1. Substituir α ↔ β por (α → β) ∧ (β → α) e substituir α → β por ¬α ∨ β

∀x ¬ (∀y ¬Animal(y) ∨ Loves(x,y)) ∨(∃y Loves(y, x))

2. Colar as negações nos átomos, utilizando as equivalências ¬(¬α) ≡ α, ¬(α ∧ β) ≡ ¬α ∨ ¬β, ¬(α ∨ β) ≡ ¬α ∧ ¬β, ¬∃x α ≡ ∀x ¬α, e ¬ ∀ x α ≡ ∃x ¬α.

∀x(∃ y¬ (¬Animal(y) ∨ Loves(x,y)))∨ (∃y Loves(y, x))

∀x(∃ y ¬¬Animal(y)∧¬Loves(x,y))∨ (∃y Loves(y, x))

∀x(∃ y Animal(y)∧¬Loves(x,y))∨ (∃y Loves(y, x))

## Transformação para Forma Clausal

∀x(∃ y Animal(y)∧¬Loves(x,y))∨ (∃y Loves(y, x))

3. Padronizar as variáveis, trocando os nomes quando estas aparecem no escopo de quantificadores diferentes

∀x(∃ y Animal(y)∧¬Loves(x,y))∨ (∃z Loves(z, x))

4. Remover os quantificadores existenciais utilizando variáveis e funções de Skolem

∀x(Animal(A)∧¬Loves(x,A))∨ Loves(B, x) (erro!)

∀x(Animal(F(x))∧¬Loves(x,F(x))) ∨ Loves(G(x), x)

5. Remover os quantificadores universais

(Animal(F(x))∧¬Loves(x,F(x))) ∨ Loves(G(x), x)

## Transformação para Forma Clausal

(Animal(F(x))∧¬Loves(x,F(x))) ∨ Loves(G(x), x)

6. Distribuir as disjunções pelas conjunções, utilizando α ∨ (β ∧ γ) ≡ (α ∨ β) ∧ (α ∨ γ)

(Animal(F(x)) ∨ Loves(G(x), x)) ∧

(¬Loves(x,F(x)) ∨ Loves(G(x), x))

Esta sentença gerou 2 cláusulas na BC

## Regras de Inferência para Cálculo de Predicados

Resolução

$p1 \vee \ldots \vee pi \vee \ldots \vee pn, m1 \vee \ldots \vee mi \vee \ldots mn \models$
$(p1 \vee \ldots \vee pi\text{-}1 \vee pi\text{+}1 \vee \ldots \vee pn \vee$
$m1 \vee \ldots \vee mi\text{-}1 \vee mi\text{+}1 \vee \ldots \vee mn)\ \sigma$

onde $pi\ \sigma = \neg mi\ \sigma$

Exemplo:

Animal(F(x)) v Loves(G(x), x) ¬Loves(u, v) v ¬Kills(u, v)

unificam com a substituição {u/G(x), v/x} e geram o resolvente:

Animal(F(x)) v ¬Kills(G(x), x)

## Exemplo: Resolução

Dado o domínio dos Deuses Gregos, considere as sentenças:

1. Todos os homens são mortais;
2. Kronos é um homem;

Provar que Kronos é mortal, utilizando refutação por resolução.

## Exemplo: Resolução

1. $\forall x$ Homem (x) → Mortal (x)
2. Homem (Kronos)

Deseja-se provar Mortal (Kronos)

Passando para a forma clausal:

| | |
|---|---|
| 1. ¬Homem (x)v Mortal (x) | Premissa |
| 2. Homem (Kronos) | Premissa |
| 3. ¬ Mortal(Kronos) | Neg. Conclusão |
| 4. Mortal (Kronos) | RES 1, 2 {x/Kronos} |
| 5. {} | RES 3, 4 |

## Exemplo: Resolução

1. Refazer o exemplo do Capitão West, utilizando resolução.
2. Seja a seguinte BC
   1. Todos que amam todos os animais são amados por alguém
   2. Qualquer um que mate um animal não é amado por ninguém
   3. Jack ama todos os animais
   4. O gato, chamado Tuna, foi morto por Jack ou por Curiosity

   Se deseja saber se Curiosity matou o gato. Resolva por refutação por resolução

## Inferência: Complexidade

O problema geral de decidir se uma base de conhecimento KB em lógica de predicados implica logicamente uma fórmula $\alpha$ **não é decidível**:

- para **alguns** conjuntos de sentenças KB, garante-se que o procedimento de prova encontra uma prova de $\alpha$ ou de $\neg\alpha$
  - problema da implicação lógica é **decidível**
- em outros casos, quando nem $\alpha$ nem $\neg\ \alpha$ são logicamente implicados por KB, o procedimento nunca termina!

## Referências Bibliográficas


- S. Russel and P. Norvig. Artificial Intelligence: A Modern Approach. Prentice Hall, Upper Saddle River, USA. 2nd. Edition, 2003. Chapter 8 and 9.
- G. Bittencourt. Inteligência Artificial: Ferramentas e Teorias. Editora da UFSC, Florianópolis. 2a. Edição, 2001. Cap. 3.